

####【置顶】相关资料与网站汇集
ChainBook/IPFS-For-Chinese
IPFS指南-知乎专栏

####【置顶】文档
Kademlia协议原理简介.pdf

####【置顶】TODO:

####【置顶】IPFS-词汇表
####【置顶】IPFS-常用通识命令
####【置顶】ipfs命令手册 / ipfs命令行简介
####【置顶】chinanf-boy/go-ipfs-zh(go-ipfs中文翻译)

###客户端模式DHT路由(--routing=dhtclient)、及其他实验阶段特性
<https://github.com/ipfs/go-ipfs/blob/master/docs/experimental-features.md>
<https://github.com/chinanf-boy/go-ipfs-zh/blob/master/docs/experimental-features.zh.md>

###storj白皮书-英文版
storj-whitepaper-v3
from: <https://github.com/storj/whitepaper>

###有关白皮书的部分解析
[Understanding the IPFS White Paper part 1](#)

IPFS世界的物流系统: libp2p - IPFS_Newb的博客 - CSDN博客

go-libp2p 入门之 PingService example

####libp2p原理及核心组件
[比肩IPFS和Filecoin，为什么说libp2p将成为区块链和物联网的基础设施？](#)
|

###有关数据存储细节
IPFS内部原理入门 (An Introduction to IPFS) -- yichigo的专栏 - CSDN博客
IPFS的存储与读取 - 区块链网

###集群搭建
IPFS系列 集群搭建 三

###有关Kademlia协议
Kademlia协议原理简介.pdf

###有关内存精简的初期调研
1.Limiting the amount of memory go-ipfs may use (<https://discuss.ipfs.io/t/limiting-the-amount-of-memory-go-ipfs-may-use/3211/3>)
1.1 ipfs本身使用只有200-300M，但由于碎片化操作系统无法回收

Quote

Jun 20

So, part of this is memory fragmentation. That is, IPFS may only be using 200-300MiB but the OS can't reclaim some of the free memory because it's mixed in with the used memory. Unfortunately, the go language makes managing memory manually rather difficult so memory fragmentation is unlikely to improve.

However, a large portion of that is probably the peerstore. We had a bug for a while where peers behind NATs were advertising every single ephemeral port they had used since they started. We've fixed that bug but still have a lot of nodes advertising these massive address lists. We're working on a couple of mitigations, but they're still in progress.

1 Like

Reply

1.2使用Swarm.ConnMgr.LowWater、Swarm.ConnMgr.HighWater限制节点维持连接数，从而节省内存。目前在mac、android电视上LowWater/HighWater 默认都是600/900。

参见config说明文档: <https://github.com/ipfs/go-ipfs/blob/master/docs/config.md#connmgr>

stebalien

Jun 18

At the moment, no. Really, go-ipfs should use no more than 50-100MiB of memory and we should have a low-
\$ Quote that uses more like 25-50MiB of memory. However, we aren't there yet.

In the meantime, you can play with the connection killing settings: Swarm.ConnMgr.LowWater and Swarm.ConnMgr.HighWater. These allow one to limit the number of connections your peer maintains and therefore the amount of memory it uses.

1.3 提到了systemd的开源软件，用于限制最大使用内存。大致看了下，systemd是要取代initd，称为linux启动的第一个进程的，使用成本会很高。

pieciar

Jun 18

Awesome info already from @stebalien , but I'll also mention that systemd.resource-control can help you tame misbehaving processes. The first thing to try is adding MemoryHigh=100M to ipfs.service ⓘ (if you don't have a ipfs.service file, start with this template ⓘ).

Also, you should probably run the service as a dedicated ipfs user, to protect your mom's account. Your own account (via ssh?) will still be able to communicate via the API endpoints on localhost.

sudo useradd ipfs --create-home
sudo -u ipfs ipfs init
sudo systemctl start ipfs@ipfs && sudo systemctl enable ipfs@ipfs
sudo loginctl enable-linger ipfs

1.4 没太懂的两个功能禁用，还不知道影响如何。

stebalien

Jun 22

You can also:

1. Start your daemon with --routing=dhtclient to disable the DHT server.
2. Not use the experimental badger datastore. It currently eats memory like nothing else (it's fast though).

1 Like

Reply

1.5 ulimit 限制进程使用系统资源的工具，超限后进程被终止。

2.Measuring memory usage #5530 (<https://github.com/ipfs/go-ipfs/issues/5530>)
ps显示使用256，pprof显示使用了76M的问题

My OS is telling me that I'm using around 256MB:

```
$ ps aux | head -n1
USER          PID    %CPU  %MEM    VSZ   RSS  TT  STAT  STARTED   TIME COMMAND
$ ps aux | grep ipfs
rob           16719    0.5   3.1 558590036 256180 s002  Ss+   2:24pm   6:28.99 ipfs daemon
$
```

But pprof tells me that it's only using 76.38MB:

```
$ curl localhost:50001/debug/pprof/heap > ipfs.heap
% Total    % Received % Xferd  Average Speed   Time    Time     Time Current
           Dload  Upload   Total     Spent    Left  Speed
100 78010   0 78010    0     0  3822k    0 --:--:-- --:--:-- --:--:-- 4009k
$ go tool pprof /Users/rob/golang/bin/ipfs ipfs.heap
File: ipfs
Type: inuse_space
Time: Sep 27, 2018 at 5:21pm (AEST)
Entering interactive mode (type "help" for commands, "o" for options)
(pprof) top5
Showing nodes accounting for 38.29MB, 50.14% of 76.38MB total
Showing top 5 nodes out of 287
      flat flat% sum%  cum  cum%
 13.47MB 17.63% 17.63%    22.97MB 30.07%  gx/ipfs/Qm0JMHf8pTrgx4E5UfM1T4Ym6kqaZeYxZJMCDX
  9.58MB 12.44% 30.07%    9.58MB 12.44%  container/List.(+List).PushFront
  6.31MB  8.26% 38.34%    6.31MB  8.26%  gx/ipfs/ondakcPmvs8yox35GqJN60f5ZQUJStHuPPTp9uX
  4.52MB  5.91% 44.25%    4.52MB  5.91%  gx/ipfs/Qncz87ZmGad78r1GfEgYjAC2yMAh9jJRD8u8Ka
  4.58MB  5.89% 50.14%    4.58MB  5.89%  gx/ipfs/QmYnsdtJ3HsodkxPE3eUj3TsCaPYvPZJ4LoXnNK
(pprof)
```

2.1内存增长的原因，碎片化，gc结束后分配内存

Stebalien commented on 29 Sep

Contributor

+ @

...

Some of it will be stacks (go routines). However, those should generally only use on the order of a few megabytes in total.

I have a sneaking suspicion that the rest is memory fragmentation. Basically:

1. We allocate something and go doesn't have space for it.
2. Go asks the kernel for a page to put this data in.
3. Go fills up this page with a bunch of random allocations.
4. We free a bunch of stuff but not *everything* in the page.

At this point, we're not using the entire page but go can't give it back to the OS. Go *will* use the rest of the space in these pages, but we may not need them (and/or the gaps in allocations are too small to be useful).

This is exacerbated by:

1. Go-ipfs's bursty allocation patterns. We'll often do a bunch of work all at once, allocate a bunch of tiny things, and then free them. Unfortunately, if we also happen to allocate a long-lived object in one of these pages full of short-lived objects, we'll have to keep the whole page around even after we free the short-lived objects. High performance systems with allocation patterns like this will actually use *separate* allocators/memory pools to keep these kinds of allocations on separate pages. Unfortunately, go doesn't really support custom allocators.

2. Garbage collection. I'm *guessing* (I haven't confirmed this) that go doesn't wait for a full GC to finish before asking the kernel to allocate a new page. That means when allocating and deallocating really *quickly*, we can end up asking the operating system for significantly more memory than we actually use at any given point in time. Unfortunately, due to (1), we may end up *having* to keep this memory reserved because we may be using small pieces of it.

TL;DR: reduce allocation frequency and use memory pools (see `Pool` in `sync`).

3.mobileapp中使用ipfs方案: IPFS in mobile use case #68 (<https://github.com/ipfs/notes/issues/68>)

Stebalien commented on 4 Apr

+ @

...

So, we've introduced a low power profile that:

1. Turns off the DHT server.
2. Severely limits the maximum number of open connections.

You can apply it with `ipfs config profile apply lowpower`.

Many of the remaining problems are architectural and will take time to fix.

在源码中搜lowpower找到相关配置内容及使用说明:

```
147 > // Lowpower: {
148 > //   Description: "Reduces daemon overhead on the system. May affect node
149 > //   functionality, performance of content discovery and data
150 > //   fetching may be degraded."
151 > ,
152 > //   Transform: func(C *Config) error {
153 > //     > c.Routing.Type = "dhtclient"
154 > //     > c.Reprovider.Interval = "0"
155 > //   },
156 > //   c.Swarm.ConnMgr.LowWater = 20
157 > //   c.Swarm.ConnMgr.HighWater = 40
158 > //   c.Swarm.ConnMgr.GracePeriod = time.Minute.String()
159 > //   return nil
160 > // },
161 > },
162 > }
[NORMAL > SPELL] +0 -0 -0 master | 1:profile.go
```

跟pc上默认配置相比，不仅降低连接节点数，延长免疫期限，type也有basic变成了dhtclient。

```
126 "AddFilters": null,
127 "ConnMgr": {
128   "GracePeriod": "28s",
129   "HighWater": 900,
130   "LowWater": 600,
131   "Type": "basic"
132 },
133 "DisableBandwidthMetrics": false,
134 "DisableNatPortMap": false,
135 "DisableRelay": false,
136 "EnableRelayHop": false
137 }
138 }
[NORMAL > SPELL] 1:config
~/ipfs/config 138L, 5296C
```

使用说明:

```
26 To better support low-power devices, we've added a low-power config profile.
27 This can be enabled when initializing a repo by running 'ipfs init' with the
28 '--profile=lowpower' flag or later by running 'ipfs config profile apply lowpower'.
29
```

4.Low Power Mode #4137 (<https://github.com/ipfs/go-ipfs/issues/4137>)

Kubuxu commented on 11 Aug 2017 • edited

Member

+ @

...

First step would be probably running go-ipfs in DHT client mode:
`ipfs daemon --routing=dhtclient`

We are working on connection closing which should further reduce the number of connections thus reducing bandwidth and power usage.

3

whyrussleeping commented on 16 Aug 2017

Member

+ @

...

Yeah, as @Kubuxu pointed out you probably want to use `--routing=dhtclient`. Its probably also worth turning off the reprovider with: `ipfs config Reprovider.Interval "0"` so your node doesn't rebroadcast all this things it has every 12 hours.

lowpower中的配置中包含了这两项配置内容。

5.Sweet-IPFS(<https://discuss.ipfs.io/t/sweet-ipfs-full-ipfs-node-for-android/3779/16>)

app代码: <https://github.com/RHazzDev/Sweet-IPFS/tree/master/app>

一个图片分享软件，使用的是kotlin语言写的，非常简洁的几个文件，可以看到使用的就是上面提到的lowpower模式的配置。

在电视上安装了apk，很多功能都未实现。

```
Thread {
    val exec = run("init")
    Thread {
        exec.inputStream.bufferedReader().forEachLine { println(it) }
    }.start()
    Thread {
        exec.errorStream.bufferedReader().forEachLine { println(it) }
    }.start()
    exec.waitFor()
    config {
        it.getAsJsonObject("Swarm").getAsJsonObject("ConnMgr").apply {
            remove("LowWater")
            addProperty("LowWater", 20)
            remove("HighWater")
            addProperty("HighWater", 40)
            remove("GracePeriod")
            addProperty("GracePeriod", "120s")
        }
    }
    progress.dismiss()
    act.runOnUiThread(callback)
}.start()
```

6.textile-go(<https://github.com/textileio/>)

一个图片分享软件，提供android app使用的aar库文件，代码: <https://github.com/textileio/textile-mobile>

7.IPFS light clients(<https://github.com/ipfs/notes/issues/206>)

iphonex上能耗问题

使用lowpower模式有所好转但仍然很高

magik6k commented on 3 Apr • edited

Member

+ @

...

@carblook, can you try running the latest go-ipfs master with low-power profile - `ipfs init --profile=lowpower` or `ipfs config profile apply lowpower` (<https://github.com/ipfs/go-ipfs/blob/master/repo/config/profile.go#L78>) and see how/if it improves things for you?

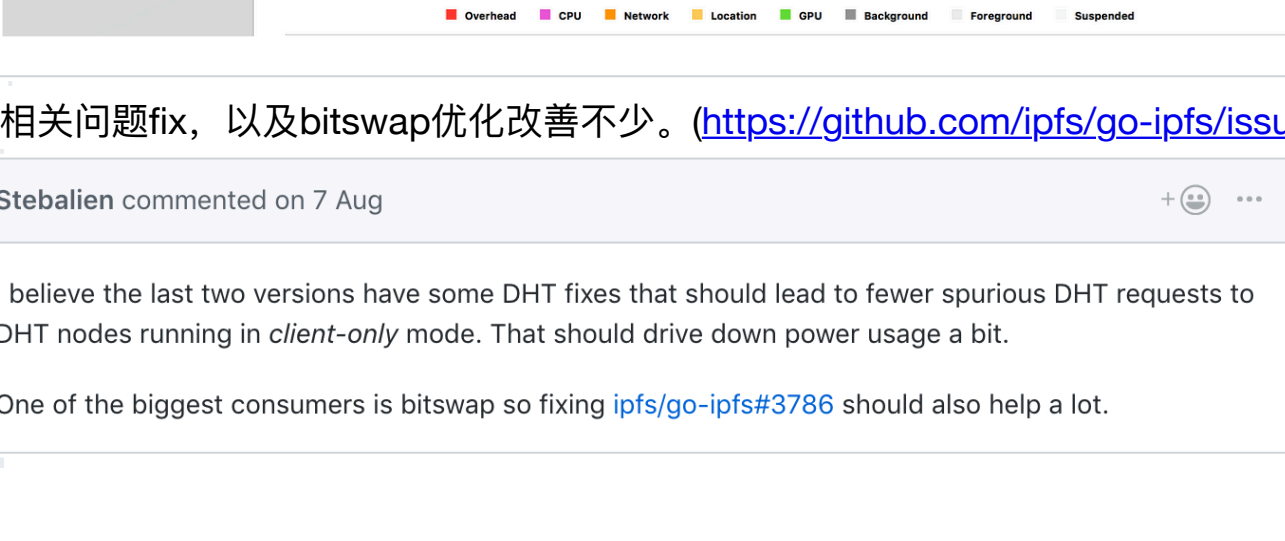
carblook commented on 11 Apr

+ @

...

Hey Lukasz,

Thank you for the suggestion. Let you know that the energy impact is improved, though still high, after applying the low-power profile.



提到有dht相关问题fix，以及bitswap优化改善不少。(<https://github.com/ipfs/go-ipfs/issues/3786>)

Stebalien commented on 7 Aug

+ @

...

I believe the last two versions have some DHT fixes that should lead to fewer spurious DHT requests to DHT nodes running in *client-only* mode. That should drive down power usage a bit.

One of the biggest consumers is bitswap so fixing [ipfs/go-ipfs#3786](#) should also help a lot.

8.The minimum requirements for the server are configured(<https://github.com/ipfs/go-ipfs/issues/4145>)

解释了--routing=dhtclient及ipfs config Reprovider.Strategy pinned会有什么影响。

还有就是ipfs/config中swarm参数配置使用，反馈是正确配置后内存使用情况好很多，但仍然在缓慢增长。

###windows平台编译/运行
[go-ipfs/docs/windows.md](#)

###有关mac、电视上环境搭建
004 IPFS - ABPS - 乐视Wiki

参考:

[\[IPFS+区块链系列\] 入门篇 - IPFS环境配置及博客搭建](#)

[打包Golang并在Android环境下运行 - CSDN博客](#)